

# Software Requirements Specification

For

Healthy Vet

Version 1.0

Prepared by

Group Name:

Lilea Anamaria Adriana CEN 4.S1B

Stan Ana-Maria Bianca CEN 4.S2B

## **TABLE OF CONTENTS**

### **1.INTRODUCTION.....**

1.1 Document Purpose

1.2 Product Scope

1.3 Definitions, Acronyms and Abbreviations

1.4 References and Acknowledgments

### **2.OVERALL DESCRIPTION.....**

2.1 Product Perspective

2.2 Product Function

2.3 End Users and Characteristics

2.4 Constrains

2.5 Operating Environment

### **3.SPECIFIC REQUIREMENTS.....**

3.1 User Interface

3.2 Functional Requirements

3.3 Performance Requirements

3.4 Design constraints

3.5 Software quality attributes

# **1 Introduction**

## **1.3 Definitions, Acronyms and Abbreviations**

Word	Definition
Microsoft Visual Studio	Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.
ASP.NET Core	ASP.NET Core is a free and open-source web framework, and higher performance than ASP.NET, developed by Microsoft and the community. It is a modular framework that runs on both the full .NET Framework, on Windows, and the cross-platform .NET Core.
framework	In computer programming, a software framework is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software.
html	HTML is the standard markup language for creating Web pages.

css	Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language like HTML
Javascript	JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm.
Bootstrap	Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.
C#	C# is a general-purpose, multi-paradigm programming language encompassing strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented, and component-oriented programming disciplines.
MySQL	an open-source relational database management system.

Word	Acronym
Cascading Style Sheets	CSS
Hypertext Markup Language	HTML
SQL	Structured Query Language

## 1.4 References and Acknowledgments

- <https://www.w3schools.com>
- <https://docs.microsoft.com/en-us/aspnet/core/fundamentals/?view=aspnetcore-3.1&tabs=windows>
- [https://gephi.org/users/gephi\\_srs\\_document.pdf](https://gephi.org/users/gephi_srs_document.pdf)
- [Bootstrap · The most popular HTML, CSS, and JS library in the world. \(getbootstrap.com\)](https://getbootstrap.com/)
- <https://stackoverflow.com/>

## 2.Overall Description

### 2.3 End Users and Characteristics

There are two types of users:

- **Pet owners**– can use the application to create an account and manage it, make appointments, delete appointments and give feedback.
- **Doctors**– can use this application to manage the appointments from pet owners, add clinic services.

This application can be used for medical purposes for pets in need of their health; their owners can make appointments, give feedback and cancel the appointment.

### 2.4 Constraints

The web application is constrained by the Internet connection is also a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.

### 2.5 Operating Environment

The developed application is compatible with computers running Windows 7 and above and can be used with Google Chrome (42 or newer), Firefox (at least 12), Microsoft Edge (12 or newer) or Opera(3.6 or newer).

Development of this application will be done using MySQL and the website will be written using HTML, CSS, C#, JavaScript, ASP.NET Core framework. For the developing environment we will use Microsoft Visual 2019.

### **3. Specific Requirements**

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features

#### **3.1 User Interfaces**

As the web application is launched, any user should see the home page that provides the landing page as seen in fig 1 – the first page contains a welcome page and 2 buttons for register – choosing one type of user: register pet owner or register doctor. Also contains images and description about the website , contains log-in button for users. Without an account, the users can see the first page , the contact page and another page with information about the website. With account for example the pet owner can see the appointment page and can add a feedback



Fig1 The Landing Page

If the user is not accessing our platform for the first time then it should click the 'Login' button from the Home page and be redirected to the User Login page. Here the user should provide only the email and password to be able to login to his/her account. A snippet for the Login page is presented in Fig 2.

If the user is using the platform for the first time then he/she should click on the button "Register now" in order to create an account for this platform. After clicking the button, he should be redirected to the Registration page. Here will be provided functionalities to require the necessary data for creating an account. The user will be asked to provide the Name, e-mail, phone number, address.

Home Info Contact

Login Register

email

confirm email

name

password

phone

Register Doctor

Creating an Account

Fig2 Create an account/Login page

The user can hold multiple appointments and he/she can also delete it or change it and the user interface should look like in fig 3. If the pet owner wants to delete the appointment, he/she needs to confirm that he wants to cancel the appointment.



HomeInfoContact

LoginRegister

Add an appointment Page

Add an appointment

Appointment

Appointment

Delete the appointment

Edit the appointment

Delete the appointment

Edit the appointment

Fig 3 Appointments page

HomeInfoContact

LoginRegister

Date and Time

Type of the appointment

Name

Phone

Submit

Adding an appointment Page

Fig 4 Add an appointments page

The user can also give feedback about the clinic in the feedback page. Also the feedback page contains images and description about the doctors that work at the vet clinic.

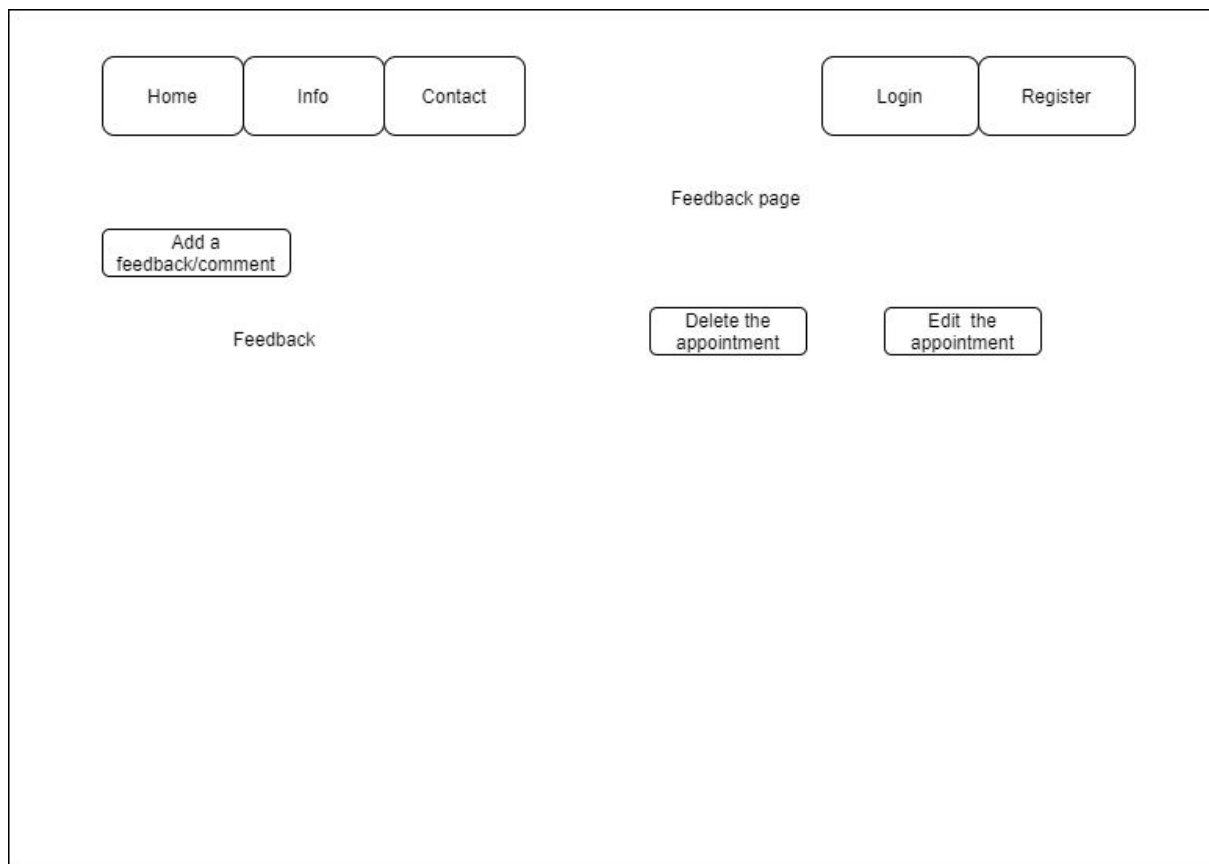


Fig4 Feedback page

### 3.1.2 Hardware Interfaces

Since the web platform does not have any designated hardware, it does not have any direct hardware interfaces. The communication between the components of the application

(database, web server and an application server) will be done by software methods, that will be operated by the underlying operating system.

- Web: Memory : 8 GB RAM, CPU : 4 core intel Xenon @ 2,2 GHz, Storage : HDD 10GB

-DataBase Server : 16 GB RAM, CPU : 4 core intel Xenon @ 2,2 GHz, Storage : HDD 100GB

### 3.1.3 Software Interfaces

#### Entity Framework

The Entity Framework is a set of technologies in ADO.NET that support the development of data-oriented software applications. The Entity Framework enables developers to work with data in the form of domain-specific objects and properties, such as customers and customer addresses, without having to concern themselves with the underlying database tables and columns where this data is stored..

More than just another object-relational mapping solution, the Entity Framework is fundamentally about enabling applications to access and change data that is represented as entities and relationships in the conceptual model. The Entity Framework uses information in the model and mapping files to translate object queries against entity types represented in the conceptual model into data source-specific queries. Query results are materialized into objects that the Entity Framework manages. The Entity Framework provides the following ways to query a conceptual model and return objects:

- LINQ to Entities. Provides Language-Integrated Query (LINQ) support for querying entity types that are defined in a conceptual model. For more information, see LINQ to Entities.
- Entity SQL. A storage-independent dialect of SQL that works directly with entities in the conceptual model and that supports Entity Data Model concepts. Entity SQL is used both with object queries and queries that are executed by using the EntityClient provider. For more information, see Entity SQL Overview.

Before integrating this framework with our platform we must start with creating the data model. Creating the data model requires to determine the main entities used in our model (eg. **Pet**

**Doctor, Appointment, Service, Feedback, Pet owner**) and establish the relationship between them (for example the relationship between Pet and Pet owner is one to many, as a pet owner can have more pets).

A set of properties must be established for each entity from our model, also the Entity Framework will recognize by default any property named "ID" as the primary key of the entity. The Entity Framework offers the possibility to set Navigation properties. Navigation

properties hold other entities that are related to a certain entity. In this case, the Pet property of a pet owner entity will hold all of the Pet entities that are related to that Pet owner entity.

### **3.1.4 Communications interfaces**

#### **HTTP/HTTPS**

HTTP is a protocol which allows the fetching of resources, such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser. A complete document is reconstructed from the different sub-documents fetched, for instance text, layout description, images, videos, scripts, and more.

Clients and servers communicate by exchanging individual messages (as opposed to a stream of data). The messages sent by the client, usually a Web browser, are called requests and the messages sent by the server as an answer are called responses.

HTTP is a client-server protocol: requests are sent by one entity, the user-agent (or a proxy on behalf of it). Most of the time the user-agent is a Web browser, but it can be anything, for example a robot that crawls the Web to populate and maintain a search engine index.

Each individual request is sent to a server, which handles it and provides an answer, called the response. Between the client and the server there are numerous entities, collectively called proxies, which perform different operations and act as caches, for example.

For us:

The client will be represented by the user-agent – in our case we have two user agents: one of them is the web browser that will make requests to the app managing server (written in .NET). This communication will be done accordingly to the HTTP protocol.

On the other side the app managing server serves as the user-agent for the database server, the app managing server will make requests to the database server.

On the opposite side of the communication channel, is the server, which serves the document as requested by the client. A server appears as only a single machine virtually: this is because it may actually be a collection of servers, sharing the load (load balancing) or a complex piece of software interrogating other computers (like DB server, or app managing server), totally or partially generating the document on demand.

### **3.3 Performance Requirements**

#### 3.3.1 View appointment.

ID: QR1

TITLE: Clear UX for viewing the appointments.

DESC: There should be a clear interface for visualizing the appointments

RAT: In order to for a user to find the appointment

DEP: none

#### 3.3.2 Finding an appointment

ID: QR2

TITLE: Easy way of find the appointment

DESC: The appointment should be easy to find through the list

RAT: In order to for a user to find the appointment

DEP: none

#### 3.3.3 Easy management of appointments

ID: QR3

TITLE: Easy management of appointments

DESC: The appointment should be easy to manage, like delete one and read its details.

RAT: In order to for the doctor to manage them.

DEP: none

#### 3.3.4 Easy way of find the services

ID: QR4

TITLE: Find the services

DESC: The services offered by the clinic should be easy to find on a separate page

RAT: In order to for a user to find the services

DEP: none

#### 3.3.5 Response time

ID: QR5

TAG: ResponseTime

GIST: The fastness of the search

SCALE: The response time of a search

MUST: No more than 5 seconds 100% of the time.

WISH: No more than 2 second 100% of the time.